

Datengetriebene Modellierung nichtlinearer Strecken mit Delaunay-Netzen

Herrn Prof. Dr. rer. nat. Dipl.-Ing. Henning Tolle zum 65. Geburtstag gewidmet

Thorsten Ullrich, Darmstadt



Dipl.-Ing. Thorsten Ullrich ist Wissenschaftlicher Mitarbeiter im Institut für Regelungstechnik, Fachgebiet Regelsystemtheorie & Robotik (Prof. Dr. rer. nat. Dipl.-Ing. H. Tolle) der TH Darmstadt. Schwerpunkte der Arbeiten: Modellierung nichtlinearer Systeme, Einsatz lernender Systeme in der Automobilelektronik.

Adresse: Technische Hochschule Darmstadt, Institut für Regelungstechnik, Fachgebiet Regelsystemtheorie & Robotik, Landgraf-Georg-Straße 4, D-64283 Darmstadt, Tel.: (061 51) 16-4990, Fax: (061 51) 16-2507, E-Mail: thul@rt.e-technik.th-darmstadt.de

Nichtlineare Regelungen erfordern den Einsatz interpolierender Elemente zur Repräsentation von Strecken- und/oder Reglerverhalten. Künstliche Neuronale Netze haben sich als prinzipiell geeigneter Ansatz hierzu erwiesen, sind aber aufgrund ihres großen Ressourcenbedarfs ungeeignet für Applikationen, in denen nur einfache Rechnersysteme eingesetzt werden können. Dies ist bei Systemen in der Kraftfahrzeugtechnik der Fall. Delaunay-Netze, die in diesem Beitrag vorgestellt werden, stellen für niedrigdimensionale Problemstellungen eine Alternative dar. Es wird ein Algorithmus zur datengetriebenen Erstellung solcher Netze angegeben und in die aktuellen Arbeiten zur Generierung problemangepasster Modellstrukturen eingeordnet. Die mit dem vorgeschlagenen Verfahren erzielten Ergebnisse werden anhand eines vereinfachten Motormomentmodells erläutert.

Data-driven modelling of nonlinear plants using Delaunay networks

Nonlinear control systems require internal representations of plant and/or controller characteristics. Basically, artificial neural networks are a promising approach for the approximation of these characteristics. However, due to their relatively high computational complexity or the required amount of storage capacity, neural networks are hardly applicable in low-cost real-time systems such as automotive control units. Delaunay networks are a computationally efficient alternative for function approximation on low-dimensional domains. The article at hand discusses this novel approach in detail and introduces an algorithm for automatic construction of these networks on the basis of measurement data. A model of steady-state engine torque characteristics is used to illustrate the proposed methods.

1 Einführung

Hohe Anforderungen hinsichtlich Sicherheit, Umweltverträglichkeit und Wirtschaftlichkeit von Produktionsanlagen und hochwertigen Konsumgütern erfordern den Einsatz moderner Verfahren der Regelungstechnik. Die zu steuernden/regelnden Strecken zeigen dabei oft stark nichtlineares und zeitvariantes Verhalten, das durch lineare parameteradaptive Algorithmen, deren Theorie ausgereift ist und die als Standardkomponenten moderner Entwicklungswerkzeuge verfügbar sind, nicht mit der geforderten Genauigkeit beherrscht werden kann. Stattdessen ist der Einsatz von Regelungskonzepten erforderlich, die auf einem nichtlinearen Prozeßmodell basieren. Zwei einschlägige Ansätze sind:

1. Der lernende Regelkreis nach Ersü [1] enthält ein neuronales Netz vom CMAC-Typ, das als prädiktives Prozeßmodell trainiert wird. Durch eine Optimierungsebene werden mithilfe dieses Modells optimale Stellaktionen generiert und in einem ebenfalls als CMAC-Netz realisierten Reglerspeicher abgelegt.
2. Ein von Narendra [2] vorgeschlagenes Konzept zur Regelung nichtlinearer Strecken basiert auf einem zeitdiskreten nichtlinearen Prozeßmodell der Form

$$y_p(k+1) = f(y_p(k)) + g(u(k)), \quad (1)$$

wobei $f(\cdot)$ und $g(\cdot)$ im allgemeinen nichtlineare Funktionen der Prozeßeingänge u und der Ausgänge y_p im vorangegangenen Abtastschritt sind. Durch die Spezifikation eines (linearen) Referenzmodells für das Führungsübertragungsverhalten

$$y_m(k+1) = \sum_{i=1}^m \alpha_i \cdot y_m(k-i) + w(k) \quad (2)$$

läßt sich der Regelalgorithmus

$$u(k) = g^{-1} \left(-f(y_p(k)) + \sum_{i=1}^m \alpha_i \cdot y_m(k-i) + w(k) \right) \quad (3)$$

ableiten. Zu dessen Anwendung müssen die Funktionen $f(\cdot)$ und $g(\cdot)$ (sowie die Inverse $g^{-1}(\cdot)$) identifiziert werden.

Diese einführenden Bemerkungen lassen erkennen, daß Systeme zur Approximation nichtlinearer Funktionen ein wesentliches Element nichtlinearer Regelungen sind und daß deren Qualität für die Funktion des Gesamtsystems entscheidend ist. Zahlreiche Arbeiten auf diesem Gebiet, beispielsweise [3; 4], schlagen die Verwendung neuronaler Funktionsapproximatoren vor. Diese Ansätze haben sich zwar im Grundsatz bewährt, nachteilig ist jedoch der hohe Ressourcenbedarf, d.h. Speicher- und Rechenaufwand, solcher Strukturen.

Gegenstand dieses Aufsatzes ist ein Konzept zur echtzeitfähigen Funktionsapproximation unter Berücksichtigung eingeschränkter Hardware-Ressourcen. Anwendungen der Kraftfahrzeugtechnik, beispielsweise Motormanagementsysteme (Einspritzung, Zündung) oder aktive Sicherheitssysteme (Anti-Blockier-System, Antriebsschlupfregelung, Fahrdynamikregelung), bilden den Applikationshintergrund der dargestellten Arbeiten. Typische Zykluszeiten dieser Systeme liegen in der Größenordnung von 10 msec.

In Kapitel 2 dieses Beitrags wird das Konzept der *Delaunay-Netze* zur Approximation nichtlinearer Funktionen eingeführt. Dieser Ansatz basiert auf aus der klassischen Mathematik bekannten Verfahren, nämlich der *Triangulierung* von Punktmengen und der *Interpolation* zwischen unregelmäßig verteilten Daten. Angewandt werden diese Methoden u. a. in der geometrischen Datenverarbeitung bei der Freiformflächenmodellierung [5], sowie im Zusammenhang mit der Methode der Finiten Elemente, etwa zur numerischen Analyse von Strömungsvorgängen [6].

Die vorliegende Arbeit zeigt die Weiterentwicklung dieser Verfahren im Hinblick auf die Approximation mehrdimensionaler Nichtlinearitäten¹ auf. Weiterhin wird ein Algorithmus zur datengetriebenen Generierung von Delaunay-Netzen angegeben (Kapitel 3). Dem Anwender wird ein Verfahren zur *automatischen Erstellung echtzeitfähiger Funktionsapproximatoren* zur Verfügung gestellt. In Kapitel 4 werden die mit dieser Methode erzielten Ergebnisse anhand simulierter Motormomentcharakteristiken diskutiert.

2 Delaunay-Netze

Das hier verfolgte Konzept zur Funktionsapproximation basiert auf der Speicherung einer Menge von Datenpunkten, die jeweils einen Schätzwert für die modellierte Funktion an einer definierten Position im Eingangsraum darstellen. Ein solcher Datenpunkt wird bei einer vektorwertigen Funktion $f(x) = (f_1(x_1, \dots, x_n), \dots, f_m(x_1, \dots, x_n))^T$ von n Veränderlichen mit m Ausgangswerten durch ein $(n + m)$ -Tupel repräsentiert:

$$s_i = (s_{i1}, \dots, s_{in}, \hat{y}_{i1}, \dots, \hat{y}_{im})^T. \quad (4)$$

s_{i1}, \dots, s_{in} definieren die *Position* des Datenpunktes s_i im Eingangsraum. Die *Attributwerte* $\hat{y}_{i1}, \dots, \hat{y}_{im}$ sind die geschätzten Ausgangswerte der modellierten Funktion f für den jeweiligen Eingangsvektor²: $\hat{y}_{ij} \approx f_j(s_{i1}, \dots, s_{in}), j = 1, \dots, m$.

Jeder Datenpunkt repräsentiert die nachgebildete Funktion f nur für einen diskreten Eingangsvektor. Um die Funktion mit einer endlichen Zahl von Datenpunkten im gesamten Eingangsraum $I \subset \mathbb{R}^n$ approximieren zu können, muß die in Form der Datenpunkte gespeicherte Information lokal verallgemeinert werden. Dies wird erreicht, indem zu beliebigen Eingangsvektoren x durch *Interpolation* zwischen im Eingangsraum nahe an x liegenden Datenpunkten Schätzwerte $\hat{f}_j, j = 1, \dots, m$ berechnet werden. Die Datenpunkte werden daher als *Interpolationsknoten* bezeichnet. Zur Vereinfachung der Schreibweise gehen die folgenden Betrachtungen von einer skalaren Funktion f aus, d. h. die Interpolationsknoten stellen $(n + 1)$ -Tupel dar, die neben der Position nur einen Attributwert enthalten.

Für die Positionen der Datenpunkte im Eingangsraum werden prinzipiell beliebige Werte zugelassen, es liegt im allgemeinen eine unregelmäßige Verteilung („scattered data“) vor. Um die Gesamtzahl N der Datenpunkte, die zur Approximation einer Funktion mit einer bestimmten Modellgenauigkeit erforderlich sind, gering zu halten, ist deren Verteilung an den lokalen *Informationsgehalt* der nachgebildeten Funktion anzupassen.

Der Informationsgehalt einer Funktion von n Veränderlichen kann formal anhand ihrer n -dimensionalen Fouriertransformierten quantifiziert werden [7]. In Bereichen des Eingangsraums, in denen hohe Ortsfrequenzen auftreten, besitzt f hohen Informationsgehalt. Dementsprechend muß die Dichte der Interpolationsknoten dort größer sein als in Gebieten niedriger Ortsfrequenzen. In Kapitel 3 wird eine heuristische Strategie zum inkrementellen Einfügen von Interpolationsknoten beschrieben, die die Anpassung an den lokalen Informationsgehalt ohne die explizite Berechnung der Fouriertransformierten gewährleistet.

Die Menge der N im Eingangsraum unregelmäßig verteilten Interpolationsknoten, beinhaltet zunächst keine topologischen Informationen; die Nachbarschaftsbeziehungen der Interpolationsknoten sowie ihre relative Lage im Eingangsraum können nicht aus der Speicherstruktur ausgelesen werden³, sondern müssen beim Vorliegen eines konkreten Eingangsvektors x explizit berechnet werden.

² Da bei realen Problemstellungen nur gestörte Meßwerte der nachzubildenden Funktion zur Verfügung stehen, handelt es sich bei den Attributen um *Schätzwerte*. Durch die in Abschnitt 3 dargestellte Optimierung werden die Attribute der N Knoten eines Delaunay-Netzes im Sinne der minimalen Fehlerquadratsumme anhand eines aus $M, M \gg N$ Punkten bestehenden Meßdatensatzes angepaßt.

³ Dies ist hingegen bei sogenannten Rasterkennfeldern der Fall, bei denen die Interpolationsknoten auf einem Gitter angeordnet sind. Rasterkennfelder stellen heute die Standardlösung in der Kraftfahrzeugtechnik dar, lassen sich aber nur für den zweidimensionalen Fall effizient realisieren und sind nur eingeschränkt für das on-line Training geeignet.

¹ Den genannten Anwendungsfeldern liegen meist zweidimensionale Netzstrukturen zugrunde.

In dem von *Militzer* [8; 9] entwickelten Interpolationsspeicherkonzept MIAS, das ebenfalls auf der Speicherung unregelmäßig verteilter Interpolationsknoten beruht, werden die euklidischen Abstände sämtlicher Knoten vom aktuellen Eingangsvektor x berechnet. Die q nächstgelegenen Knoten werden zur Bestimmung des Ausgangsschätzwertes herangezogen. Dieses Vorgehen besitzt jedoch zwei Nachteile. Zum einen führt das reine Abstastkriterium in ungünstigen Fällen zu Extrapolationen und somit zu verminderter Modellgüte [10]. Außerdem ist die Berechnung und Sortierung der Abstände zu allen Interpolationsknoten mit einem erheblichen Rechenaufwand verbunden und damit für die Anwendung in Echtzeitsystemen in der Kraftfahrzeugtechnik ungeeignet. In [11] wurden Verfahren vorgestellt, die durch eine Segmentierung des Eingangsraums die Auswahl der Interpolationsknoten vereinfachen. Dennoch liefern auch diese Ansätze noch keine befriedigenden Antwortzeiten und führen in einigen Fällen ebenfalls zu einer suboptimalen Auswahl der Interpolationsknoten.

Das Konzept der Delaunay-Netze begegnet dieser Problematik durch die Speicherung expliziter topologischer Informationen. Dazu wird eine *Triangulierung* der Interpolationsknoten im n -dimensionalen Eingangsraum vorgenommen, d.h. jeweils $n+1$ Knoten werden zu n -dimensionalen Simplexes (Dreiecke im zweidimensionalen Fall, Hypertetraeder im allgemeinen Fall) verkettet. Die Attributwerte der $n+1$ Interpolationsknoten eines Simplex definieren eine n -dimensionale Hyperebene, die als Modell der approximierten Funktion innerhalb des Gültigkeitsgebietes dieses Simplex angenommen wird (lineare Interpolation). Als Gültigkeitsgebiete werden die konvexen Hüllen der Simplexes definiert. Bild 1a veranschaulicht dieses Konzept für den zweidimensionalen Fall.

Die folgenden Abschnitte behandeln zunächst die Frage, wie die unregelmäßig verteilten Interpolations-

knoten zu n -dimensionalen Simplexes verkettet werden, und wie für einen gegebenen Eingangsvektor effizient auf die aktiven Knoten zugegriffen werden kann.

2.1 Delaunay-Triangulierung

Aus einer Menge von N Knoten an beliebigen Positionen $x_i \in \mathbb{R}^n$ lassen sich grundsätzlich

$$N_{t_{\max}} = \binom{N}{n+1} = \frac{N!}{(n+1)! \cdot (N-n-1)!} \quad (5)$$

n -dimensionale Simplexes bilden. Die zur Funktionsapproximation geeigneten N_t Simplexes müssen jedoch zwei wesentliche Bedingungen erfüllen:

1. Überschneidungsfreiheit:

Die Schnittmenge zweier Simplexes T_i, T_j ist leer: $T_i \cap T_j = \{\}, i, j \in \{1, \dots, N_t\}, i \neq j$.

2. Eingangsraumüberdeckung:

Die Vereinigungsmenge sämtlicher N_t Simplexes ist gerade gleich dem Definitionsbereich der nachgebildeten Funktion:

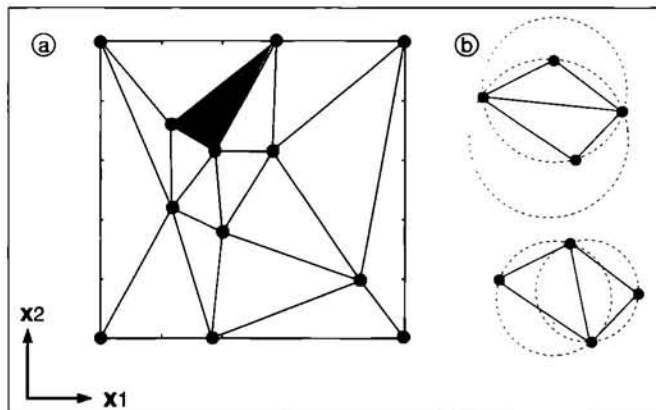
$$\bigcup_{k=1}^{N_t} T_k = I \subset \mathbb{R}^n.$$

Die erste Bedingung stellt die Eindeutigkeit des Ausgangsschätzwertes sicher. Würden sich die Gültigkeitsgebiete zweier Interpolationsknotengruppen überschneiden, wäre die Auswahl der aktiven Knoten mehrdeutig. Die Bedingung der vollständigen Eingangsraumüberdeckung dient der Vermeidung von Extrapolationen, die im allgemeinen mit ungenauen Schätzwerten verbunden sind und durch die zusätzlich erforderlichen Berechnungen das Antwortzeitverhalten ungünstig beeinflussen würden. Praktisch kann diese Bedingung dadurch erfüllt werden, daß stets 2^n Interpolationsknoten auf den „Ecken“ des n -dimensionalen Definitionsbereichs positioniert werden.

Ein Hilfsmittel aus der geometrischen Datenverarbeitung, die *Delaunay-Triangulierung*, ermöglicht die Bestimmung einer Menge von Simplexes, die die beiden oben genannten Bedingungen erfüllen. Delaunay-Simplexes besitzen die Eigenschaft, daß ihre n -dimensionalen Umkugeln keinen weiteren Knoten des Gesamtnetzes umfassen [5]. Anhand einer Anordnung von vier Knoten in der Ebene ($n=2$) verdeutlicht Bild 1b die Delaunay-Definition. Bei der oben im Bild dargestellten Triangulierung umfassen die Umkreise der beiden Dreiecke jeweils den vierten Knoten. Das untere Bild stellt hingegen die Delaunay-Triangulierung dieser Knoten dar.

Die Delaunay-Triangulierung hat gegenüber anderen Triangulierungsformen den Vorteil, daß die Simplexes tatsächlich überschneidungsfrei sind, vorausgesetzt die Knoten befinden sich in allgemeiner Lage⁴, die durch kleine pseudo-zufälligen Verschiebungen der Knoten stets sichergestellt werden kann.

Bild 1: a) Beispiel einer Triangulierung von Interpolationsknoten (ausgefüllte Punkte) im zweidimensionalen Eingangsraum. Der durch ein Kreuz markierte Eingangsvektor liegt im Gültigkeitsgebiet des schattierten Dreiecks. Dementsprechend werden die Attributwerte der Interpolationsknoten, die zu diesem Dreieck gehören (aktive Knoten), zur Schätzung des Funktionswertes durch lineare Interpolation herangezogen. Die Attributwerte stellen die „Höhen“ der nachgebildeten Funktion über der (x_1, x_2) -Ebene dar. b) Zur Delaunay-Definition.



⁴ Es darf keine Untermenge von $n+1$ Knoten mit kollinearen Positionsvektoren und keine Untermenge von $n+2$ Knoten, die auf dem Rand einer n -dimensionalen Hyperkugel liegen, existieren.

Weiterhin ist die Delaunay-Triangulierung in besonderer Weise zur Funktionsapproximation durch lineare Interpolation innerhalb ihrer Simplices geeignet. Für den zweidimensionalen Fall läßt sich zeigen, daß durch Verwendung von Delaunay-Simplices die Sobolev-Seminorm, die ein Maß für die „Rauhigkeit“ der nachgebildeten Fläche darstellt, minimiert wird [12]. Omohundro [13] leitet aus den Eigenschaften der zweidimensionalen Delaunay-Triangulierung noch eine weitergehende Aussage ab: Unter der wenig einschränkenden Bedingung endlicher Krümmungen der nachzubildenden Funktion liefert die lineare Interpolation mithilfe der Delaunay-Simplices den minimalen Maximalfehler.

Diese Zusammenhänge sind zunächst erstaunlich; die Delaunay-Triangulierung minimiert einen Güteindex, der von der nachgebildeten Funktion abhängt, obwohl diese Funktion bei der Berechnung der Triangulierung überhaupt nicht berücksichtigt wird⁵. Eine anschauliche Begründung für die positiven Interpolationseigenschaften der Delaunay-Triangulierung ist die *Kompaktheit* der Delaunay-Simplices. Die Delaunay-Definition vermeidet die Entstehung langer und schmaler Simplices⁶. Diese würden dazu führen, daß zur Ausgangsgrößenschätzung teilweise sehr weit vom aktuellen Eingangsvektor entfernt liegende Interpolationsknoten herangezogen werden. Formal lassen sich diese Eigenschaften der Delaunay-Triangulierung nur für den zweidimensionalen Fall ableiten, in [12] wird jedoch zumindest für dreidimensionale Delaunay-Triangulierungen die Aussage der Kompaktheit auf Delaunay-Tetraeder übertragen. Für die im Hinblick auf industrielle Applikationen – beispielsweise in der Kraftfahrzeugtechnik – relevanten zwei- bis vierdimensionalen Eingangsräume wird daher die Delaunay-Triangulierung als optimale Struktur angesehen.

Zusätzlich zu den Positionen und Attributwerten der Interpolationsknoten müssen deren topologische Relationen (Triangulierung) gespeichert werden. Hierzu sind verschiedene Datenstrukturen bekannt [14]. Für die vorliegende Anwendung zur schnellen Funktionsapproximation ist ein Kompromiß zwischen Speicherbedarf und algorithmischem Aufwand beim Zugriff auf die Struktur anzustreben. Dies leistet die von Lawson [15] vorgeschlagene Datenstruktur, die zu jedem Simplex $2 \cdot (n + 1)$ Einträge enthält; dies sind die Indices der $n + 1$ Interpolationsknoten, die den Simplex bilden, sowie die Indices der $n + 1$ an den Randecken des Simplex angrenzenden Nachbarsimplices. Die Information über die Nachbarsimplices ermöglicht unter der Voraussetzung begrenzter Änderungsgeschwindigkeit des Eingangsvektors eine sehr schnelle Lokalisierung der aktiven Knoten.

Insgesamt ist die Darstellung der Topologieinformation mit einem vergleichsweise hohen Speicheraufwand

verbunden. Ein aktuell untersuchter Ansatz zur Reduktion des Speicherbedarfs besteht darin, nur für eine geeignet gewählte Teilmenge der Interpolationsknoten topologische Informationen zu speichern und die Feinstruktur in der Umgebung des aktuellen Eingangsvektors jeweils zu rekonstruieren.

2.2 Auswahl der aktiven Knoten

Die Auswahl der aktiven Knoten zu einem Eingangsvektor x entspricht der Bestimmung des Simplex, dessen konvexe Hülle x enthält. Dieser kann effizient anhand der *baryzentrischen Koordinaten* $b_i(x)$ relativ zu einzelnen Simplices der Struktur ermittelt werden. Definitionsgemäß sind die baryzentrischen Koordinaten eines Punktes x in Relation zu einem Simplex T , der durch die Interpolationsknoten s_1, \dots, s_{n+1} gebildet wird, diejenigen $n + 1$ Gewichtungskoeffizienten, die den Schwerpunkt von T in den Punkt x verschieben. Da diese Definition nur n Gleichungen für die $n + 1$ unbekannten b_i liefert, wird die Summe der baryzentrischen Koordinaten auf 1 normiert.

$$\sum_{i=1}^{n+1} b_i(x) \cdot (s_{i1}, \dots, s_{in})^T = x \quad \sum_{i=1}^{n+1} b_i(x) = 1 \quad (6)$$

Somit erhält man ein lineares Gleichungssystem, dessen Lösung die baryzentrischen Koordinaten liefert. Zur Auswahl der aktiven Knoten in einem Delaunay-Netz kann die folgende Eigenschaft der baryzentrischen Koordinaten ausgenutzt werden:

Befindet sich ein Punkt $x \in \mathbb{R}^n$ innerhalb der konvexen Hülle eines n -dimensionalen Simplex T , so sind sämtliche $n + 1$ baryzentrische Koordinaten b_i , $i = 1, \dots, n + 1$ von x in Relation zu T nichtnegativ. Befindet sich x außerhalb der konvexen Hülle von T , ist mindestens eine baryzentrische Koordinate negativ [16].

Aufgrund der Überschneidungsfreiheit der Simplices existiert stets genau ein Simplex, der $n + 1$ nichtnegative baryzentrische Koordinaten liefert. Die Trivalllösung zur Bestimmung dieses Simplex besteht im linearen Durchsuchen der Simplex-Liste (Lawson-Datenstruktur). Im statistischen Mittel müssen dann 50 % aller Simplices untersucht werden, so daß sich keine akzeptablen Antwortzeiten einstellen. Eine drastische Beschleunigung läßt sich jedoch erreichen, wenn die Tatsache ausgenutzt wird, daß sich der Eingangsvektor x im aktuellen Abtastzyklus in der Umgebung seiner vorherigen Position aufhalten wird. Dies ist damit zu begründen, daß x mit dem Zustands- oder Steuervektor eines dynamischen Systems in Verbindung steht (siehe etwa Gleichung (3)) und sich somit bei realen Systemen nicht schlagartig ändern wird, sondern kontinuierlichen Trajektorien folgt.

Daher ist es sinnvoll, die Suche der aktiven Knoten von dem im vorangegangenen Zyklus selektierten Simplex T_{prev} zu beginnen. Zunächst werden die baryzentrischen Koordinaten des aktuellen Eingangsvektors x in Relation zu T_{prev} berechnet. Ergibt sich hierbei eine negative baryzentrische Koordinate, wird die Suche bei einem der Nachbarsimplices von T_{prev} , die in der Lawson-Datenstruktur eingetragen sind, fortge-

⁵ Die Triangulierung hängt ausschließlich von den Positionen der Interpolationsknoten und nicht von den zugeordneten Attributwerten (Funktionswerten) ab.

⁶ Im zweidimensionalen Fall ist das Delaunay-Kriterium identisch mit der Maximierung des kleinsten Dreieckswinkels [5; 14].

setzt. Die Datenpakete sind so organisiert, daß aus dem Index einer negativen baryzentrischen Koordinate auf den Nachbarsimplex, der den Eingangsvektor mit hoher Wahrscheinlichkeit enthält, geschlossen werden kann. Ergeben sich dennoch auch für den ausgewählten Nachbarn negative baryzentrische Koordinaten, so wird die Suche entsprechend fortgesetzt. Die beschriebene Strategie wird in ähnlicher Weise in [17] skizziert und dort als *orientierte Laufsuche* bezeichnet. Untersuchungen mit realen, in einem Kraftfahrzeug gemessenen Eingangssignalen (Drosselklappenwinkel und Drehzahl) haben gezeigt, daß durch dieses Verfahren im Mittel sehr wenige Simplices betrachtet werden müssen. Die daraus resultierenden Antwortzeiten werden in Kapitel 4 erläutert.

3 Datengetriebene Generierung von Delaunay-Netzen

Zur Ableitung eines Verfahrens zur datengetriebenen Generierung von Delaunay-Netzen wird zunächst eine mathematische Formulierung des Ein-Ausgangsverhaltens angegeben⁷. Der Ausgangsschätzwert \hat{y} zu einem Eingangsvektor x ergibt sich aus dem inneren Produkt eines von x abhängigen Gewichtsvektors w und dem Vektor der N Attributwerte der Interpolationsknoten $\hat{y} = (\hat{y}_1, \dots, \hat{y}_N)^T$. Somit gilt

$$\hat{y}(x) = w(x)^T \cdot \hat{y}. \quad (7)$$

Der Gewichtsvektor $w(x)$ enthält an den Positionen, die den $n + 1$ aktiven Knoten entsprechen, die zugehörigen baryzentrischen Koordinaten b_i und an den verbleibenden $N - n - 1$ Positionen jeweils den Wert 0. Die Gewichtung der Attributwerte der aktiven Knoten mit den baryzentrischen Koordinaten entspricht einer linearen Interpolation im n -dimensionalen Raum⁸. Die restlichen 0-Elemente von w tragen dem lokalen Charakter des Interpolationsverfahrens Rechnung.

Gleichung (7) macht deutlich, daß das Ein-Ausgangsverhalten eines Delaunay-Netzes durch zwei sequentielle Abbildungen beschrieben werden kann. Zuerst wird der n -dimensionale Eingangsvektor x auf einen N -dimensionalen Gewichtsvektor w abgebildet. Dies stellt eine nichtlineare Operation dar; es findet eine Auswahl von $n + 1$ aus N Knoten statt, deren Gewichte nichtlinear von den Positionsvektoren abhängen (Lösung des Gleichungssystems (6) durch Berechnung von Determinanten). Dieser nichtlinearen Operation nachgeschaltet ist eine lineare Abbildung des N -dimensionalen Gewichtsvektors w auf den m -dimensionalen (in Gleichung (7) skalaren) Ausgang. Das Verhalten eines Delaunay-Netzes ist also nichtlinear bezüglich der Positionen und linear bezüglich der Attribute der Interpolationsknoten.

Ein effizientes Verfahren zur Erstellung von Delaunay-Netzen anhand von Meßdaten wird daher eine heuristische Strategie zur Positionierung der Interpolationsknoten mit einer linearen Optimierung der Attributwerte verbinden. Sind insgesamt M Meßdatensätze gegeben, wobei die Zahl der Meßdatensätze im allgemeinen sehr viel größer als die Zahl der Interpolationsknoten sein wird, erhält man das folgende überbestimmte Gleichungssystem:

$$y_{\text{mes},k} = w(x_{\text{mes},k})^T \cdot \hat{y}, \quad k = 1, \dots, M \quad (8)$$

Hierin bezeichnet $x_{\text{mes},k}$ die Position des k -ten Meßpunktes im Eingangsraum des Delaunay-Netzes und $y_{\text{mes},k}$ den entsprechenden Meßwert, d. h. den Wert der durch das Netz nachzubildenden Funktion an dieser Position. Faßt man die Gewichtsvektoren w , die sich an den M Meßpunktpositionen ergeben, in einer Meßmatrix W und die M Meßwerte in einem Vektor y_{mes} zusammen, so lassen sich die im Sinne des least-squares Kriteriums optimalen Attributwerte durch Pseudoinversion von W bestimmen:

$$\hat{y}_{\text{opt}} = (W^T W)^{-1} W^T \cdot y_{\text{mes}} \quad (9)$$

Die Attributoptimierung kann also in geschlossener Form stattfinden und braucht nicht durch ein langwieriges iteratives Training vorgenommen zu werden. Als numerisch robustes Verfahren zur Berechnung von (9) eignet sich beispielsweise die Methode der Singulärwertzerlegung (singular value decomposition, [18]).

Zur heuristischen Lösung des nichtlinearen Optimierungsproblems der Positionierung der Interpolationsknoten kommen grundsätzlich drei Vorgehensweisen in Betracht. *Konstruktive* Verfahren gehen von einem kleinen Initialnetz aus, in das schrittweise zusätzliche Knoten eingefügt werden. *Destruktive* Ansätze starten mit einem komplexen Netz, aus dem schrittweise die Knoten mit dem geringsten Informationsgehalt entfernt werden⁹. *Evolutionäre Algorithmen* stellen eine Mischform aus konstruktiven und destruktiven Verfahren dar. Ausgehend von einer Population von Netzen werden durch Mutation und (im Falle der Genetischen Algorithmen) Rekombination Netze sowohl mit verminderter als auch vergrößerter Knotenzahl generiert.

Zur Erstellung von Delaunay-Netzen wurde der konstruktive Ansatz gewählt, da hierbei der geringste Gesamtrechenaufwand zu erwarten ist. Die bei der Netzgenerierung (off-line Phase) benötigte Rechenzeit spielt zwar eine untergeordnete Rolle, trotzdem ist es anzustreben, dem Anwender ein Verfahren, das interaktives Arbeiten ermöglicht, zur Verfügung zu stellen.

Als heuristisches Kriterium zum Einfügen von Knoten wird der Meßpunkt, an dem das momentan bestehende Netz den maximalen Modellfehler aufweist, herangezogen. Die Position des eingefügten Knotens wird gleich dieser Meßpunktposition gewählt und der Attributwert wird mit dem entsprechenden Meßwert initia-

⁷ Der im folgenden beschriebene Algorithmus wird *off-line* abgearbeitet. Hierbei spielen Rechenaufwand und Speicherbedarf eine untergeordnete Rolle.

⁸ Der Beweis dieser Aussage ist im Anhang skizziert.

⁹ Diese Strategie ist im Zusammenhang mit neuronalen Netzen unter dem Begriff „pruning“ bekannt.

liert. Nach jedem eingefügten Knoten werden die Attribute aller bis zu diesem Zeitpunkt vorhandenen Knoten durch LS-Optimierung, d. h. Anwendung von (9), anhand des gesamten Meßdatensatzes angepaßt.

Die einzige Interaktion des Anwenders besteht darin, ein Kriterium für den Abbruch der Netzgenerierung zu spezifizieren. Dies kann in Form der mindestens erforderlichen Modellgüte oder der höchstens zulässigen Zahl von Interpolationsknoten erfolgen¹⁰.

Das Verfahren läßt sich wie folgt zusammenfassen:

- I:** Definition eines Abbruchkriteriums, d. h. der geforderten Modellgenauigkeit oder der erlaubten Modellkomplexität (Knotenzahl).
- II:** Definition eines initialen Satzes von (mindestens) $N = 2^n$ Knoten, die die Ecken des gewünschten Definitionsbereichs besetzen und somit Extrapolationen verhindern.
- III:** Attributoptimierung für das bestehende Netz durch Berechnung der Meßmatrix W und Anwendung von (9).
- IV:** Hinzufügen eines Knotens:
 - (a) Bestimmen der absoluten Modellfehler für sämtliche M Meßpunkte:

$$e(x_{\text{mes},k}) = |y_{\text{mes},k} - \hat{y}(x_{\text{mes},k})|, \quad k = 1, \dots, M$$
 - (b) Bestimmen der Meßpunktposition, an der das Netz die maximale Abweichung vom Sollverhalten produziert:

$$x_{\text{mes},j} = \arg \max(e(x_{\text{mes},k})), \quad k = 1, \dots, M$$
 - (c) Einfügen eines Knotens an der Position $x_{\text{mes},j}$ mit dem Attributwert $y_{\text{mes},j}$:

$$s_{N+1} = (x_{\text{mes},j}, y_{\text{mes},j})^T \quad N \equiv N + 1$$
- V:** Falls das Abbruchkriterium noch nicht erfüllt ist, Wiederholung ab Schritt **III**.

Bezüglich der Kombination von Heuristiken zur Strukturoptimierung und mathematischer Optimierung der Attribute besitzt dieses Verfahren Parallelen zu der Erstellung von neuronalen Netzen durch genetische Algorithmen [19]. Dort wird die Netztopologie heuristisch, durch Mutation und Rekombination, und die Gewichte des Netzes durch ein Standardlernverfahren (Error-Backpropagation) bestimmt. Auch die von Marenbach und Bettenhausen [20] entwickelten Verfahren zur Erzeugung strukturierter Modelle nichtlinearer dynamischer Systeme mit genetischen Algorithmen basieren auf einer solchen Kombination. Die Struktur der Modelle wird durch den genetischen Algorithmus festgelegt, die Parameter der Teilmodellblöcke durch Optimierung mittels Hooke-Jeeves-Verfahren bestimmt. Diese kombinierten Strategien haben sich als sehr leistungsfähig erwiesen und bewähren sich auch bei der automatischen Erstellung von Delaunay-Netzen.

¹⁰ Dadurch kann bei der Netzgenerierung die auf dem Zielsystem verfügbare Speicherkapazität berücksichtigt werden.

4 Ergebnisse

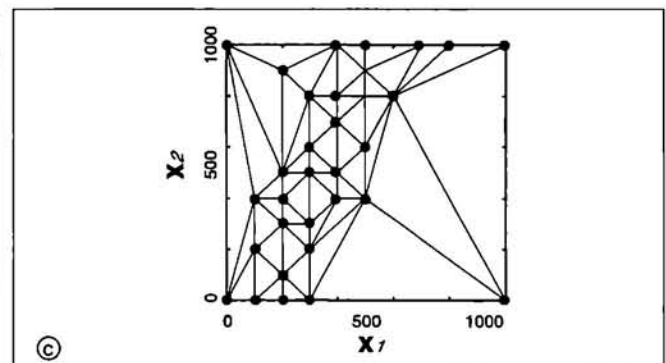
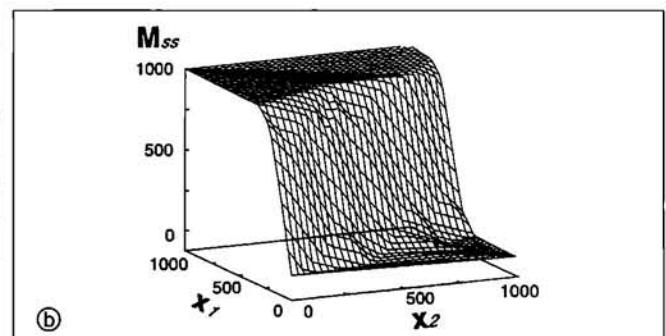
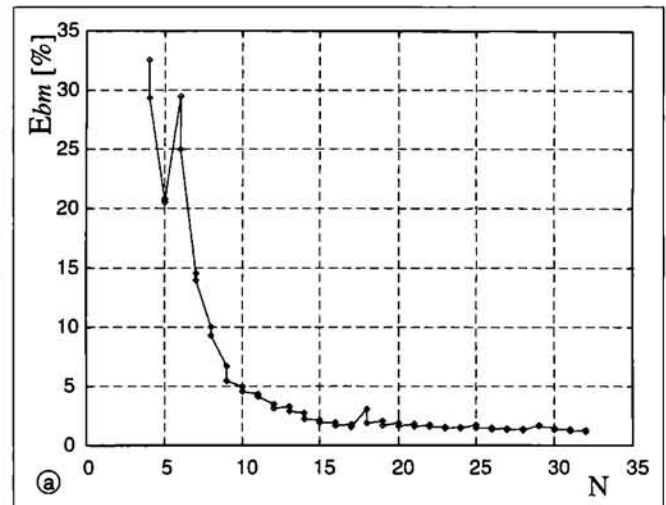
Eine typische Anwendung nichtlinearer Modelle in der Kraftfahrzeugtechnik ist die Nachbildung der Drehmomentcharakteristik eines Verbrennungsmotors. Haupteinflußgrößen sind hierbei die Drosselklappenstellung und die Motordrehzahl. Zur Durchführung systematischer Untersuchungen wird diese Charakteristik durch den funktionalen Zusammenhang

$$y(x_1, x_2) = 1000 \cdot [1 + e^{(2 - \frac{1}{10} x_1 + \frac{1}{100} x_2)}]^{-1}, \quad x_1, x_2 \in [0, 1000] \quad (10)$$

simuliert [21]. Das im stationären Betrieb abgegebene Drehmoment entspricht der Ausgangsgröße y , die Ein-

Bild 2: Generierung eines stationären Motormomentmodells.

- a) Entwicklung des Modellfehlers über der Knotenzahl N
- b) Resultierende Ein-Ausgangs-Charakteristik des generierten Netzes
- c) Verteilung und Triangulierung der Interpolationsknoten in der (x_1, x_2) -Ebene



gänge Drosselklappenwinkel und Drehzahl entsprechen x_1 und x_2 . Als Grundlage der Netzgenerierung dienten 121 auf Gitterpunkten in der (x_1, x_2) -Ebene berechnete „Meßpunkte“. Das Startnetz enthielt vier Knoten, die die Eckpunkte des jeweils durch die Intervalle $[0, 1000]$ gegebenen Definitionsbereichs besetzen. Als Abbruchkriterium wurde eine Knotenzahl von $N = 32$ vorgegeben. Bild 2a zeigt den Verlauf des bezogenen mittleren absoluten Fehlers E_{bm} während der Netzgenerierung.

Zu jedem Wert von N enthält das Diagramm 2a zwei Fehlerwerte, nämlich den Modellfehler nach dem Hinzufügen des jeweils letzten Knotens und den Fehler nach der anschließend ausgeführten Attributoptimierung. Man erkennt, daß insbesondere zu Beginn des Prozesses eine deutliche Reduktion des Modellfehlers durch den Optimierungsschritt erreicht wird¹¹.

Bild 2b zeigt die durch das generierte Netz repräsentierte Ein-Ausgangs-Charakteristik, 2c die Verteilung der Interpolationsknoten sowie deren Delaunay-Triangulierung. Das heuristische Kriterium für das Einfügen der Knoten bewirkt tatsächlich eine Anpassung der Knotendichte an den lokalen Informationsgehalt der nachgebildeten Funktion; im Bereich starker Krümmungen (hohe Ortsfrequenzen) werden deutlich mehr Knoten angeordnet als in den äußeren, näherungsweise linearen Bereichen.

Zur grundsätzlichen Validierung des Konzepts der Delaunay-Netze bei der Nachbildung mehrdimensionaler Nichtlinearitäten wird die durch Gleichung (10) beschriebene Motormomentcharakteristik auf 3 und 4 Eingänge erweitert:

$$y(x_1, \dots, x_n) = 1000 \cdot [1 + e^{(2 - \frac{1}{30} x_1 + \frac{1}{100} \sum_{i=2}^n x_i)}]^{-1}, \quad x_i \in [0, 1000], \quad n = 3, 4 \quad (11)$$

Hierbei wird unterstellt, daß sich weitere Eingangsgrößen in derselben Weise wie die Drehzahl (x_2) auf das Motormoment auswirken würden. Als Grundlage der datengetriebenen Netzgenerierung wurden wiederum auf einem Raster im gesamten Definitionsbereich berechnete „Meßpunkte“ herangezogen. In Bild 3 sind die Verläufe des bezogenen mittleren absoluten Fehlers E_{bm} während der Netzgenerierung aufgetragen. Als Initialnetze wurden bei diesen Untersuchungen jeweils die 2ⁿ zur Vermeidung von Extrapolationen notwendigen Eckpunkte vorgegeben. Das Erreichen einer Knotenzahl von $N = 48$ im dreidimensionalen und $N = 80$ im vierdimensionalen Fall diente als Abbruchkriterium.

Hinsichtlich des bezogenen mittleren absoluten Fehlers führen diese Knotenzahlen nicht zu einer vergleichbaren Modellgenauigkeit ($E_{bm} = 4,9\%$ bei $n = 3$, $E_{bm} = 12,7\%$ bei $n = 4$). Hierbei muß allerdings berücksichtigt werden, daß der Güteindex E_{bm} die Nach-

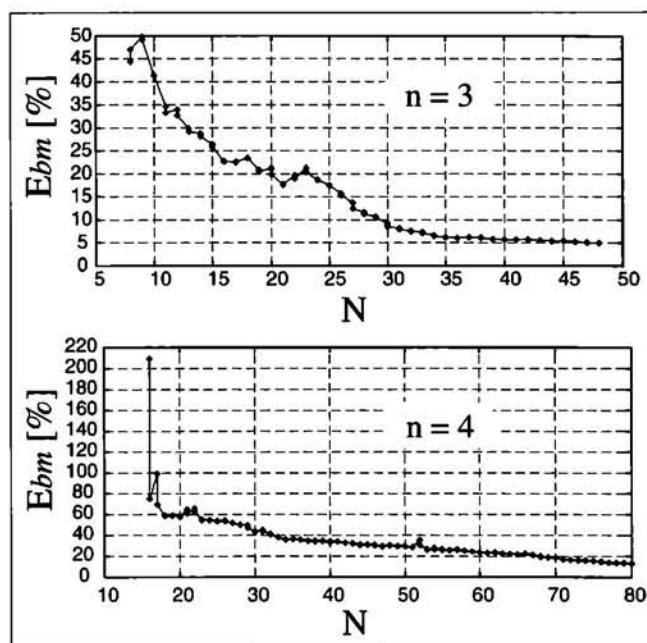


Bild 3: Zur Generierung 3- und 4-dimensionaler Modelle.

Tabelle 1: Leistungsmerkmale der automatisch generierten Delaunay-Netze.

N	E_{bm} [%]	Speicherbedarf [byte]	T_{avg} [μ sec]	T_{max} [μ sec]
2				
32	≤ 1	96 + 79	41	1383
3				
48	4,9	192 + 514	65	4248
4				
80	12,7	400 + 2400	363	20418

bildungsgenauigkeit im gesamten Definitionsbereich bewertet. Beim Einsatz im Regelkreis werden hingegen nur Trajektorien im Eingangsraum durchlaufen, so daß die mehrdimensionale Nichtlinearität nur in einem kleinen Teilgebiet des Eingangsraums tatsächlich exakt nachgebildet werden muß.

Tabelle 1 faßt die Leistungsmerkmale der generierten Netze zusammen. Der Speicherbedarf setzt sich aus dem für die Interpolationsknoten benötigten Speicherplatz (erster Summand) und dem Speicheraufwand zur Darstellung der topologischen Relationen (Delaunay-Triangulierung) zusammen. Bei der Ermittlung von Speicheraufwand und Antwortzeiten wurde eine aufwandsoptimierte Variante der Delaunay-Netze zugrunde gelegt. Hierbei wird lediglich eine Grobtopologie gespeichert, d.h. nur eine Teilmenge der Interpolationsknoten in die Triangulierung einbezogen¹². Die exakte Delaunay-Triangulierung (Feinstruktur) wird jeweils lokal in der Umgebung der aktuellen Eingangsvektoren x rekonstruiert.

Die in Tabelle 1 angegebenen Rechenzeiten wurden unter Verwendung eines T805-Mikroprozessors mit 30 MHz Taktfrequenz ermittelt. Die mittleren Ant-

¹¹ Die vereinzelt zu beobachtende Verschlechterung ist darauf zurückzuführen, daß die bei der Netzgenerierung minimierten Gütekriterien (max. absoluter Fehler beim Einfügen von Knoten, Summe der Fehlerquadrate bei der Attributoptimierung) nicht zwangsläufig zu einer Reduktion des in den Diagrammen aufgetragenen Güteindex E_{bm} führen.

¹² 50% aller Knoten im zwei- und dreidimensionalen, 40% im vierdimensionalen Fall.

wortzeiten (T_{avg}) sind sehr gering, während in ungünstigen Situationen sehr viel längere Rechenzeiten auftreten (T_{max}). Diese „ungünstigen Situationen“ sind gekennzeichnet durch hohe Änderungsgeschwindigkeiten der Eingangssignale¹³. Die orientierte Laufsuche benötigt in diesen Fällen sehr viel mehr Schritte als in quasistationären Betriebsphasen, in denen häufig derselbe oder zumindest unmittelbar benachbarte Simplexes aktiviert werden. Falls die auftretenden maximalen Antwortzeiten die Echtzeitbedingungen einer konkreten Applikation verletzen, sind einfache „Ausweichstrategien“ anzuwenden. Beispielsweise kann bei Überschreiten der maximal zulässigen Antwortzeit zunächst der letzte gültige Ausgangswert beibehalten werden. Wenige Zyklen später wurde mit hoher Wahrscheinlichkeit ein geeigneter Simplex bestimmt, von dem ausgehend wieder sehr schnelle Berechnungen möglich sind.

5 Schlußbemerkung

Das Konzept der Delaunay-Netze hat sich im Grundsatz zur Realisierung echtzeitfähiger Funktionsapproximatoren bewährt. Durch den vorgestellten Algorithmus zur Netzgenerierung entstehen problemangepaßte Netze, deren Interpolationsknotenverteilung den lokalen Informationsgehalt der nachzubildenden mehrdimensionalen Funktion berücksichtigt. Die wesentliche Beschränkung dieses Konzeptes liegt in der relativ geringen Zahl von Eingangsgrößen; die diskutierten Ergebnisse zeigen, daß Delaunay-Netze mit mehr als 4 Eingängen wiederum mit einem erheblichen Ressourcenbedarf verbunden wären. Für die vorgesehenen Applikationen in der Kraftfahrzeugtechnik sind drei- und vierdimensionale Modelle, die sich effizient mit dem vorgeschlagenen Konzept realisieren lassen, jedoch sicher ausreichend und stellen eine wesentliche Erweiterung der heute eingesetzten zweidimensionalen Rasterkennfelder dar.

Die weiteren Arbeiten auf dem Gebiet der Delaunay-Netze werden sich insbesondere mit dem on-line Training zur Kompensation zeitvarianter Prozeßcharakteristiken befassen. Neben der Korrektur der Attributwerte der Interpolationsknoten erlaubt das verfolgte Konzept auch die Modifikation der Interpolationsknotenverteilung. Ein weiterer Schwerpunkt der Arbeiten ist die Modellierung nichtlinearer dynamischer Prozesse durch Delaunay-Netze. Neben der direkten Erfassung des Ein-Ausgangsverhaltens [22] besteht ein aussichtsreicher Ansatz in der Verwendung parametrischer Prozeßmodelle, deren Parameter arbeitspunktabhängig von einem Delaunay-Netz bereitgestellt werden.

Danksagung

Der Autor bedankt sich bei der Deutschen Forschungsgemeinschaft (DFG), die das vorgestellte Projekt finanziert, sowie bei Dr. Thomas Roos, ETH Zürich, für die wertvollen Diskussionen zum Thema der Delaunay-Triangulierungen.

¹³ In den diskutierten Beispielen verursacht durch abruptes Verstellen der Drosselklappe.

Anhang

Zur Linearität der Interpolationsvorschrift

Aus (6) läßt sich die folgende Darstellung der baryzentrischen Koordinaten gewinnen (Lösung des linearen Gleichungssystems):

$$b_i(x) = c_{0i} + \sum_{j=1}^n c_{ji} \cdot x_j, \quad i = 1, \dots, n+1 \quad (12)$$

Man beachte, daß die Koeffizienten c_{ji} unabhängig von den aktuellen Eingängen $x = (x_1, \dots, x_n)^T$ sind. Die durch (7) gegebene Interpolationsvorschrift läßt sich schreiben als

$$\hat{y}(x) = \sum_{i=1}^{n+1} b_i(x) \cdot \hat{y}_i, \quad (13)$$

wobei \hat{y}_i die Attribute der $n+1$ aktiven Knoten bezeichnen. Einsetzen von (12) in (13) liefert

$$\begin{aligned} \hat{y}(x) &= \sum_{i=1}^{n+1} c_{0i} \hat{y}_i + \sum_{i=1}^{n+1} \sum_{j=1}^n c_{ji} x_j \hat{y}_i = \\ &= \underbrace{\sum_{i=1}^{n+1} c_{0i} \hat{y}_i}_{K_0} + \sum_{j=1}^n x_j \underbrace{\sum_{i=1}^{n+1} c_{ji} \hat{y}_i}_{K_j} \end{aligned}$$

Damit ist eine Darstellung der Netzantwort als *Linearkombination* der Attribute der aktiven Knoten gefunden.

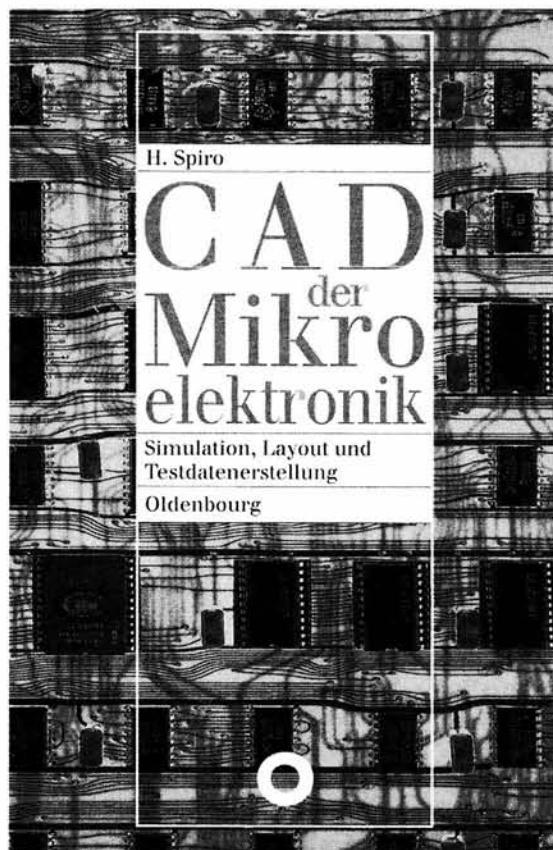
Literatur

- [1] Tolle, H. und Ersü, E.: Neurocontrol. Learning Control Systems Inspired by Neuronal Architectures and Human Problem Solving Strategies. In Thoma, M. (Hrsg.), Lecture Notes in Control and Information Sciences No. 172. Springer, Berlin, Heidelberg, New York etc. 1992.
- [2] Narendra, K.S. und Parthasarathy, K.: Identification and Control of Dynamical Systems Using Neural Networks. IEEE Transactions on Neural Networks 1 (1), S. 4–27, 1990.
- [3] Schiffmann, W. und Geffers, H.: Adaptive Control of Dynamic Systems by Backpropagation Networks. Neural Networks 6 (4), S. 517–5424, 1993.
- [4] Gehlen, S., Hormel, M. und Kopeck, J.: Einsatz Neuronaler Netze zur Kontrolle komplexer industrieller Prozesse. at – Automatisierungstechnik (2), S. 85–91, Februar 1995.
- [5] Hoschek, J. und Lasser, D.: Grundlagen der geometrischen Datenverarbeitung. B.G. Teubner, Stuttgart 1989.
- [6] Irmisch, S. und Schwolow, R.: Erzeugung unstrukturierter Dreiecksnetze mittels der Delaunay-Triangulierung. Zeitschrift für Flugwissenschaft und Weltraumforschung 18, S. 361–368, 1994.
- [7] Sanner, R.M. und Slotine, J.-J.E.: Function Approximation, „Neural“ Networks and Adaptive Nonlinear Control. In 3rd IEEE Conference on Control Application 1994.
- [8] Tolle, H., Militzer, J. und Ersü, E.: Zur Leistungsfähigkeit lokal verallgemeinernder Speicher und ihren Einsatzmöglichkeiten in lernenden Regelungen. messen – steuern – regeln msr (32/3), S. 98–105, 1989.
- [9] Tolle, H. und Militzer, J.: Vergleich von Speicherkonzepten für lernende Regelungen. In Fachtagung Automatisierung, Dresden, GDR 1988.
- [10] Gehlen, S.: Ein Expertensystem zur Führung komplexer Prozesse mit integrierten lernfähigen Speichersystemen. In VDI-Berichte Nr. 897: VDI/VDE-GMA-Aussprachetag „Wissensverarbeitung in der Automatisierungstechnik“, Düsseldorf, Langen, FRG 1991, VDI-Verlag.
- [11] Schmitt, M., Ullrich, T. und Tolle, H.: Associative Datafields in Automotive Control. In 3rd IEEE Conference on Control Applications, S. 1239–1244, Glasgow (UK) 1994.

- [12] Aurenhammer, F.: Voronoi-Diagrams – A Survey of a Fundamental Geometric Data Structure. ACM Computing Surveys 23, S. 345–405, September 1991.
- [13] Omohundro, S. M.: The Delaunay Triangulation and Function Learning. Technischer Bericht 90-001, International Computer Science Institute, Berkeley, CA 1989.
- [14] Cline, A. K. und Renka, R. L.: A Storage-Efficient Method For Construction Of A Thiessen Triangulation. Rocky Mountain Journal of Mathematics 14, S. 119–139, 1984.
- [15] Lawson, C. L.: Software for C^1 Surface Interpolation, S. 161–194, Academic Press 1977.
- [16] Barnhill, R. E.: Representation And Approximation Of Surfaces, S. 82–120, Academic Press 1977.
- [17] Palacios-Velez, O. und Renaud, B. C.: A Dynamic Hierarchical Subdivision Algorithm for Computing Delaunay-Triangulations And Other Closest-Point Problems. ACM Transactions on Mathematical Software 16, S. 275–292, September 1990.
- [18] Press, W. H., Flannery, B. P., Teukolsky, S. und Vetterling, W. T.: Numerical Recipes in C – The Art of Scientific Computing. Cambridge University Press, 2nd, Auflage 1989.
- [19] Schiffmann, W.: Evolutionäres Design von Neuronalen Netzen. In Hofestädt, R., Krückeberg, F. und Lengauer, T. (Hrsg.), Informatik in den Biowissenschaften, S. 121–132, 1993.
- [20] Marenbach, P., Bettenhausen, K. D. und Cuno, B.: Selbstorganisierende Generierung strukturierter Prozeßmodelle. at – Automatisierungstechnik 43 (6), S. 227–288, 1995.
- [21] Schmitt, M.: Untersuchungen zur Realisierung mehrdimensionaler lernender Kennfelder in Großserien-Steuergeräten. PhD thesis, Technische Hochschule Darmstadt, VDI-Verlag-Fortschrittsberichte Reihe 12 Nr. 246, 1995.
- [22] Ullrich, T. und Tolle, H.: Delaunay Networks for Modelling of Non-Linear Processes. In IASTED/ISMM International Conference Modelling and Simulation, S. 319–322, Pittsburgh, USA, April 1996.

Manuskripteingang: 1. März 1996.

Verfügbar unter
lediglich die vom Gesetz vorgesehenen Nutzungsrechte gemäß UrhG



Hans Spiro

CAD der Mikroelektronik

Simulation, Layout und
Testdatenerstellung

1997. 426 Seiten. 284 Abbildungen
52 Tabellen. Zahlreiche Aufgaben mit
Lösungen

DM 78,-

ISBN 3-486-24114-1

Die Komplexität moderner Mikroelektronik verlangt unbedingt den Einsatz von Computerprogrammen zur Entwicklung mikroelektronischer Schaltungen. CAD-Programme wie SPICE, CHARLES u. a. werden zur Simulation der zu entwickelnden Schaltungen, andere CAD-Programme zur Konstruktion der Hardware, dem sogenannten Layout von Chips und Platinen, oder zur Erzeugung von Testdaten eingesetzt. Das vorliegende Buch führt in die verschiedenen Methoden und Verfahren der Simulation, der Layoutgestaltung und der Testdatenerstellung ein und erläutert die hierfür am häufigsten ein-

gesetzten Algorithmen durch exemplarische Beispiele und ausgewählte Aufgaben mit Lösungen.

Neu ist das Hauptanliegen des Titels, den Ingenieur, der mit der Entwicklung mikroelektronischer Schaltungen oder Systeme betraut ist, zum entscheiden den Systemdenken hinzuführen, ihm somit den Überblick über alle Bereiche der Mikroelektronikentwicklung zu verschaffen und ihm die große Bedeutung von CAD in der Mikroelektronik zu vermitteln.

Aus dem Inhalt:

Die CAD-gestützte Mikroelektronik-Entwicklung – Simulationsebenen – Schaltkreissimulation – Prozeß- und Bauelemente- (Device-) Simulation – Logiksimulation – Register-Transfer-Simulation – High-Level Simulation – Struktur und Aufbau integrierter Schaltungen – Layout auf der Siliziumebene, interne Zellenkonstruktion – Ablauf eines mit CAD-Unterstützung durchgeführten Designs – Partitionierung – Platzierung – Verdrahtung – Entwurfskontrolle – Testdatenerstellung

In Ihrer Buchhandlung oder direkt beim Verlag:

R. Oldenbourg Verlag · Rosenheimer Straße 145 · 81671 München

Internet: <http://www.oldenbourg.de>

FAX 089/45051-204